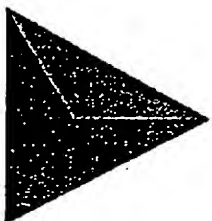


# Exhibit B

# **Efficient Mixing of Sequential Prefetches With Random Access Data in a Preexisting LRU Cache**

**Patent Review Board**

**IBM Confidential**

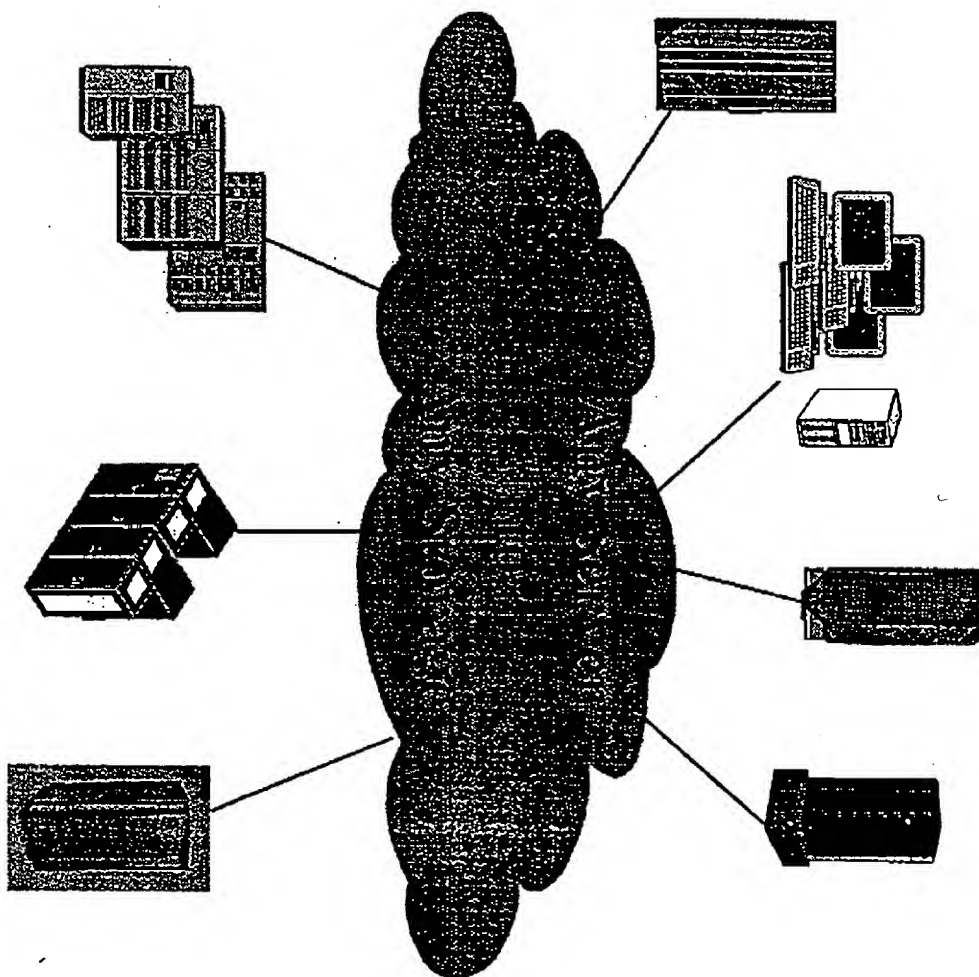


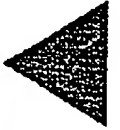
**IBM Confidential**

## **Efficient Prefetching w/LRU...**

- Title:
  - ▶ Efficient Mixing of Sequential Prefetches with Random Access Data in a Preexisting LRU Cache.
- Product Name:
  - none (several candidates: CF, SAN)
  - ▶ Has not been disclosed.
- First Date Disclosed:
  - ▶
- Problem Solved:
  - Prefetch optimization in a black-box LRU caching system.

# Example use of the process in a SAN environment





**IBM Confidential**

## **Problem**

- LRU cache logic sometimes inaccessible or undesirable to alter.
- LRU cache often accessed by external hosts or multiple hosts.
- Prefetched data is ideal for sequential accesses, which works poorly with LRU.
- Identifying prefetch candidates difficult.



# Solution

---

- Solution is to:
  - ▶ Use model of prefetching effects to make decisions.
  - Estimate single ref. residency time.

**IBM Confidential**

## **Efficient Prefetching w/LRU...**

- Benefit over prior art:
  - Competitors:
    - ▶ Improved identification of prefetch candidates.
    - Modification of LRU not required.
  - Companies trying to preload selected elements in an LRU cache.
- Will others want to use it?
  - Consider a SAN with a cached control unit to preload into.
- Discoverable?
  - ▶ I/O requests in addition to workload together with a moving prefetch criteria.
- ▶ how easily?
- Alternatives?
  - Static prefetch determination.



**IBM Confidential**

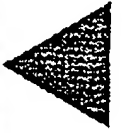
## **The process for efficient prefetching w/LRU**

- Determine cache size.
- Periodically fetch Hit Ratio, cached I/O rate and estimate the SRRP.
- For each I/O check the model's buffer, use previous element's sequential count.
- If above a dynamic threshold, prefetch.



## **The process for efficient prefetching w/LRU (continued)**

- Load I/O requests + prefetches into model's buffer. Update LRU position on hit. Discard on overflow.
- Keep track of overflow point for each of multiple prefetch threshold criteria, *misses* counting prefetched "hits" for each criteria.
- If alternate prefetch criteria value is better, then consider adjustment.



**IBM Confidential**

## **Description of Invention**

### ■ Inputs:

- ▶ Cache Size
- ▶ periodic:
  - Hit Ratio feedback
  - Cached I/O Rate feedback

### ■ Output:

- ▶ Prefetch candidates identified
- ▶ Optimal prefetch threshold identified
- ▶ Value of prefetching quantifiable

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☒ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**